Performance of Classifier Architectures With the RNADS Feature Space

Mark Wellman
U.S. Army Research Laboratory
Adelphi, MD 20783-1197

Abstract

To evaluate the efficiency of the remote netted acoustic/seismic sensor array (RNADS) [1–6] for classification, we must investigate the performance of various classification algorithms. Currently, the U.S. Army Research Laboratory (ARL) is developing an acoustic/seismic target classifier using a backpropagation neural network (BPNN) algorithm. Various techniques for extracting features have been evaluated to improve the confidence level and probability of correct identification (ID) [1,3,6]. For any given feature space, the BPNN creates complex boundaries in the hyperspace occupied by the feature vectors and only one hidden layer is required to create hyperplanes as decision boundaries [7,8]; this, however, may not be the ideal classifier. Alternately, nonparametric and parametric classifier architectures are being investigated, since it is the mutual relationship between features and classifiers that allows the maximum recognition performance. Intuitively, we expect the BPNN to perform well, based on results from k-means analysis techniques. Using a hierarchical k-means analysis tool, we determined that only a few feature data clusters exist for each class. These "feature pockets" may comprise about 40 percent of the training data in some instances and, in fact, have been suggested to be useful in a minimum distance classifier or beneficial in learning vector quantization. The nonparametric classifier architectures make no assumption about the statistics of the feature space distribution [7] and instead, rely on the data to estimate classification parameters. They have advantages when the features are created using nonlinear processes with highly non-Gaussian statistics and allow flexibility in the tradeoff of computation, memory, training, and testing. In this report, we present results using the BPNN classifier, a nearest cluster classifier (NC), a simplified form of the k-nearest neighbor algorithm [9–12], and radial basis functions (RBF), which is a neural network architecture where the hidden units provide a set of functions that constitute an arbitrary basis for the input patterns. We will also present several parametric classifier results. These include a linear regression classifier (LIN), which forms a linear mapping between the output (class) variable and the input variables (features); the logistic regression classifier, which uses the logistic function in the mapping; and various multivariate Gaussian classifiers.

1 Introduction

We discuss the research at ARL in the classification of ground vehicles, based on the fusion of acoustic and seismic collocated sensors. A fundamental problem we have addressed before [1–6,13] is the selection of robust features that are stable and class specific. At this point, the harmonic line association features (HLA) and seismic shape statistics have proven to be the best choice [14]. Further improvements are under investigation using propagation models and harmonic tracking techniques [15] that will alleviate some of the problems associated with the

Report Documentation Page

Report Date 00 Sep 1999	**Report Type** N/A	**Dates Covered (from... to)** -

Title and Subtitle Performance of Classifier Architectures With the RNADS Feature Space	**Contract Number**
	Grant Number
	Program Element Number
Author(s)	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) U.S. Army Research Laboratory Adelphi, MD 20783-1197	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es) Department of the Army, CECOM RDEC Night Vision & Electronic Sensors Directorate AMSEL-RD-NV-D 10221 Burbeck Road Ft. Belvoir, VA 22060-5806	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)

Distribution/Availability Statement
Approved for public release, distribution unlimited

Supplementary Notes
See also ADM201471, Papers from the Meeting of the MSS Specialty Group on Battlefield Acoustic and Seismic Sensing, Magnetic and Electric Field Sensors (2001) Held in Applied Physics Lab, Johns Hopkins Univ, Laurel, MD. on 24-26 Oct. 2001. Volume 2 (Also includes 1999 and 2000 Meetings)

Abstract

Subject Terms

Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU

Number of Pages
10

nonstationary nature of acoustic and seismic signatures [5,7]. An issue that we have not thoroughly addressed and comprises the bulk of this report is the performance of differing classifier topologies. It is certain that the Bayes likelihood ratio test is optimal in minimizing the classification error, but in most examples, one must estimate the feature space density functions using a finite number of samples. The estimation can be arduous and often requires a large number of samples to be accurate. Instead of using this approach, we simply looked at the performance of several readily available architectures, both nonparmetric and parametric.

2 Parametric and Nonparametric Classifier Architectures

The parametric classifiers perform quite well when the statistical parameters of the model fit the underlying multidimensional probability distribution. However, the opposite is often true and one can expect very poor results when the mismatch is too great between the true statistical parameters and the assumed ones that ultimately govern class separability and the model. The LIN classifier generates a multivariate linear relationship between the input variables and the output or class variable. For example,

$$y = w_0 + w_1 x_1 + \ldots.. + w_\mathrm{n} x_\mathrm{n} + \mathrm{error} \tag{1}$$

is the defining relationship with the solution resulting from the minimization of the sum of squared errors over the data.

The solution of the least-squares problem determines the weights, resulting in the following:

$$W^* = \left(X^T X\right)^{-1} \left(X^T D\right) \tag{2}$$

with W^* being a column vector of the weights and

$$X = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \ldots & \ldots \\ 1 & X_N \end{bmatrix}, \qquad D = \begin{bmatrix} d_1 \\ d_2 \\ \ldots \\ d_N \end{bmatrix}, \tag{3}$$

with X_i being the i-th input vector and d_i the desired output for the i-th input vector of the training set.

Linear regression is used as a baseline algorithm to compare all other classification algorithms. When the actual relationship is nonlinear, LIN generates a very poor model.

As mentioned, the logistic regression classifier generates a map between the input variables and the output according to the logistic function

$$y = \frac{1}{\left(1 + e^{-\Sigma}\right)} \tag{4}$$

with

$$\Sigma = w_0 + \sum_{i=1}^{N_\mathrm{input}} w_i x_i \ . \tag{5}$$

See references [11,20] for further information concerning the training of the logistic regression architecture; for convenience, some of the important relationships for training are given below in equations (6) and (7).

To fit the logistic function to the data-set, the cross entropy cost function is commonly used.

$$E_k = d_k \bullet \ln\left(1/y_k\right) + \left(1 - d_k\right) \bullet \ln\left(1/\left(1 - y_k\right)\right) \ . \tag{6}$$

Here, E_k is the error from the i-th sample pattern, y_k is the output produced with the input vector of the k-th example pattern (x_k), and d_k is the desired output for the k-th example pattern. The ln is the natural logarithm.

The computation of the gradient of the cross entropy error summed over the entire training set allows the weight update, which results in the following:

$$\Delta w_i = \eta x_{ki}\left(d_k - y_k\right) \tag{7}$$

where η is the learning rate for gradient descent and d_k is the desired class output for the k-th pattern presented. Stopping criteria can be number of epochs, maximum time for training, or rms error for classification.

A standard parametric classifier is the multivariate Gaussian classifier, which makes the assumption that the underlying feature distribution is multivariate normal and characterizes each class by a mean vector and a covariance matrix.

$$p\left(X/c_i\right) = \frac{1}{\left(2\pi\right)^{M/2}\left|R_i\right|^{1/2}} \exp\left(-\frac{1}{2}\left(X - \mu_i\right)R_i^{-1}\left(X - \mu_i\right)\right) \tag{8}$$

M = feature dimension
μ^i = i-th class mean vector
R_i = i-th class covariance matrix

Classification of the unknown feature vector X is accomplished by computing the conditional probability $p(X/c_i)$ for all classes, the prior probability $p(c_i)$ by simple frequency techniques, and then calculating the likelihood ration given by

$$L_j\left(X\right) = p\left(X/c_j\right)p\left(c_j\right) \ , \tag{9}$$

where the X is assigned to the class j with the maximum L_j.

This technique will not perform as well as nonparametric techniques if the data distribution is non-Gaussian, but it serves as a useful benchmark for more complex algorithms.

The Gaussian mixture classifier is a refinement of the previous technique. This is actually considered a nonparametric technique since we are using the data to generate the mixed Gaussians. In this case, the probability distributions are modeled as a weighted set of Gaussians. Here, the conditional distributions are given by

$$p\left(X \ / \ c_j\right) = \sum_{k=1}^{Ng} w_k G_k \ , \tag{10}$$

where w_k is the weight of the k-th Gaussian G_k and the weights sum to one. The Gaussians are given by

$$G_k = \frac{1}{(2\pi)^{n/2}\left|R_k\right|^{1/2}} \exp\left(-\frac{1}{2}\left(X - M_k\right)^T R_k^{-1}\left(X - m_k\right)\right) . \tag{11}$$

As before, R_k is the covariance matrix for the k-th Gaussian and M_k is the corresponding mean vector. Training is accomplished using an iterative procedure known as the estimate-maximize algorithm [16], which maximizes the likelihood of the training set generated by the probability distribution function. The parameters estimated are simply the weights, mean vectors, and covariance. See the references [9,10] for the training procedure.

Another nonparametric classifier we considered is the nearest cluster (NC) approach. This algorithm is very similar to the *k*-nearest neighbor (kNN) technique but is a simplification of it, requiring less memory and computation time. The NC classifier first performs k-means analysis on the training data to form a set of clusters; with each of these clusters, we can calculate a posterior probability for each class represented in each cluster.

For example, the posterior probability would be expressed as

$$p\left(\omega_j / N_k\right) = \frac{N_{jk}}{N_k} \ , \tag{12}$$

with

$\omega_j =$ class j,
$N_{jk} =$ number of tokens from class j within the k-th cluster, and
$N_k =$ number of tokens in k-th cluster.

Equation (12) is a simple frequency method for determining probabilities.

Classification occurs by first determining the closest cluster to the unknown feature vector using a Euclidean metric; the unknown is then classified according to the largest posterior probability in that cluster. Performance of this classifier is suboptimal but useful, since in some instances, it can perform as well as the kNN classifier.

The third nonparametric classifier we considered is the RBF neural network. This classifier can be described mathematically by

$$p\left(y / c_j\right) = \sum_{k=1}^{N_c} y_k \varphi\left(\left\| y / c_j \right\|\right) \ , \tag{13}$$

with:

$N_c =$ number of clusters,
$C_k =$ k-th cluster vector,
$\lambda_k =$ weight of k-th cluster centroid,
$\| \quad \| =$ euclidean norm, and
$\phi(\ ) =$ nonlinear transformation function.

The nonlinear transformation function can take several forms. The choice is important only in a few instances and is not considered crucial to the RBF performance [17]. In this paper, we used the Gaussian as the nonlinear transformation function. The technique employed for the

RBF classifier is a special case of the linear regression model [18] that exploits orthogonal least-squares learning. Typical characteristics for the RBF classifier include a moderate training time, fast classification, and moderate memory requirements.

3 Boundary Decision Architectures

Several boundary decision architectures are available; for example, the simplest and most often used is the BPNN, which allows for separation of complex hyperboundaries, depending on the number and size of the hidden layers. The fast BPNN is an improvement over the standard BPNN because of its judicious selection of initial weights and faster convergence. Some studies have shown superior recognition performance as well [7,12]. The discriminant neural network is noteworthy because its successive approximations of class boundaries via linear discriminant functions maximize separability [19]. There are others as well that represent slight improvements in convergence and/or recognition, but the tradeoff is in the complexity of the algorithm and its implementation.

Through *k*-means analysis, we have seen that feature sets exhibit only a few clusters with minor overlap; in this case, a boundary decision classifier like the BPNN should perform well. We have used the BPNN with an adaptive learning rate that allows fine-grain adjustments during training. Smoothing is also incorporated and allows the control of weight adjustment, based on past values of gradient descent, and can prevent the training process from terminating in a shallow local minimum. The hyperbolic tangent is used instead of the sigmoid function because its zero-centered transform reduces training time.

$$y = \frac{\left(1 - e^{-\text{net}}\right)}{\left(1 + e^{-\text{net}}\right)} \tag{14}$$

with

$$\text{net} = w_0 + \sum_i w_i x_i \ . \tag{15}$$

For target ID, the artificial neural network (ANN) can provide both a robust classifier and a measure of confidence in the classification decision. The ANNs derive their computational power from the parallel-distributed structure and the ability to learn and adapt.

4 RNADS Feature Space

The nonstationary characteristics of the acoustic signatures increase the difficulty in the feature selection process, but methods to alleviate this difficulty somewhat have been reported [1,3]; the most notable to date is the use of the HLA algorithm with higher order shape statistics [6].

The HLA algorithm takes advantage of spectral characteristics that are dominated by narrow-band spectral peaks. In the past, the narrow-band spectral peaks were used for classification purposes, either in hierarchical clustering schemes or as direct inputs into an ANN. The spectral peaks are typically bandlimited between 1 and 400 Hz, but peak components occur between 10 and 120 Hz. The majority of tracked and wheeled vehicles of interest are diesel powered and, thus, the engine firing rate and track slap produce these spectral peaks. When considering feature methods based solely on the acoustic spectrum, the entire set of spectral peaks can often be used. Alternately, one can select specific frequency components to improve signal-to-noise ratio (SNR) by using split-window peak picking and/or HLA, each of which has the added benefit of appreciably reducing the feature space while maintaining class separability.

The HLA technique selects those peaks that are harmonically related to create harmonic line sets for each frame (frame = 1 second) of data samples. After split-window peak picking is performed on the PSE feature set, the algorithm finds the maximum peak in the frequency set and assumes that this peak is some *k*-th harmonic line of the fundamental frequency subject to the following soft constraint for fundamental frequency range,

$$f_{\text{fund}} \in \left\{8{,}20\right\} \text{Hz} \ , \tag{16}$$

and then calculates the total signal strength in this HLA set. The integer value *k* that gives the maximum signal strength is assumed to be the correct harmonic line number and the harmonic lines of this particular set are retained as a feature vector. This technique has two advantages: (1) the feature vector is normalized and is based solely on harmonic line number and not a function of frequency, and (2) the peak energy is tracked frame to frame, thus producing a predominance pattern for the acoustic energy source [3].

The naval community has found shape statistic features (SSF) to be beneficial in certain classification problems. The SSFs have been used in evaluating the discrepancy between the correlated and uncorrelated components of return energy in low-frequency active target-echo characterization [7]. Also, shape transition statistics have found use in discriminating biologic from manmade sounds by exploiting minute differences in broadband energy [7]. When one looks carefully at the spectrograms for the seismic data, it is apparent that shape statistics could exploit some of the differences in tracked versus wheeled spectrums and, perhaps, the small changes in the spectral content of various tracked vehicles. Based on these findings, we have investigated the use of the higher order shape statistics as well as introducing a temporal shape transition statistic for classifying the seismic feature sets. A useful temporal shape transition statistic is simply the absolute change in the shape mean for each subsequent frame. See references [7,13,14] for further information concerning the RNADS shape statistic features.

5 Data Collection and Classifiers

We gathered the combined acoustic/seismic data from collocated acoustic and seismic sensors using a three-axis seismic sensor configured as part of an acoustic sensor array that ARL uses on typical field experiments. HLA and shape statistic features data [13,14] are extracted from the set of vehicles and then split into a testing and training file. The training file will typically consist of 70 percent of the whole data-set for the classifiers. For each classifier architecture, we performed several train/test experiments to achieve optimization. Crossvalidation was then performed using vehicle data that was withheld from the original train/test procedure. All the odd class labels used in this experiment are heavy-tracked vehicles with the exception of class 7, a light-tracked vehicle. All the even classes are very heavy or heavy-wheeled vehicles. Classes 1 to 4 have similar origins, as do classes 5 to 7.

6 Results

To qualify the classification performance, a confusion matrix is calculated that provides the percentage of correct identification (C_{ID}) for each class of ground vehicles based on the following expression:

$$C_{ID} = \frac{NP_{CID}}{K} \quad , \tag{17}$$

with K being the total number of observations and $N_{P_{CID}}$, the number of correct decisions.

The rms error, a measure of the confidence in the classifier and the overall classification error for each target, was calculated by the following:

$$E_{rms} = \sum_{i=1}^{N_{\text{train}}} \left(\frac{1}{2} \sum_{j=1}^{J} \left(d_{ij} - y_{ij} \right)^2 \right) \quad , \tag{18}$$

where i is indexing each training input pattern and j is indexing each output class. The E_{rms} for training and crossvalidation was statistically no different. A simple percentage relationship for all test samples is given by

$$Error_{\text{class}} = 100 \times \frac{N_{\text{misclassed}}}{N_{\text{total test tokens}}} \quad , \tag{19}$$

which is a useful indicator of overall performance.

The following tables are the crossvalidations using the HLA and the SSF feature space derived for the various targets. The values in the tables are the mean score for several train/test and crossvalidation runs with each classifier architecture. The numbers represent percentage of correct identifications as expressed in equation 18.

Table 1. Linear Regression

Actual	1	2	3	4	5	6	7
1	78	0	7	0	2	6	5
2	2	73	0	22	7	8	0
3	9	3	85	1	15	3	7
4	0	17	0	68	11	0	0
5	0	7	0	7	47	22	5
6	1	0	0	0	10	47	2
7	10	0	8	3	8	14	81

$E_{rms} = 0.285$ $Error_{\text{class}} = 32.7\%$

Table 2. Unimodal Gaussian

Actual	1	2	3	4	5	6	7
1	75	1	7	0	3	6	4
2	0	61	0	8	0	2	0
3	10	1	74	0	1	2	0
4	0	8	0	71	3	0	0
5	8	29	14	21	83	57	44
6	1	0	0	0	5	33	4
7	6	0	5	0	5	0	48

$E_{rms} = 0.377$ $Error_{\text{class}} = 33.5\%$

Table 3. Logistic Regression

Actual	1	2	3	4	5	6	7
1	85	2	4	0	7	2	3
2	0	81	0	2	1	0	2
3	10	2	96	2	5	0	6
4	0	9	0	89	11	0	0
5	1	6	0	7	57	2	1
6	1	0	0	0	12	81	0
7	3	0	0	0	7	15	88

$E_{rms} = 0.248$ $Error_{\text{class}} = 22\%$

Table 4. Gaussian Mixture

Actual	1	2	3	4	5	6	7
1	89	14	6	4	4	2	9
2	0	76	0	2	2	2	0
3	4	1	79	1	6	1	2
4	0	6	1	88	13	0	3
5	4	3	3	5	70	0	6
6	0	0	0	0	2	79	2
7	3	0	11	0	3	16	78

$E_{rms} = 0.377$ $Error_{\text{class}} = 19.8\%$

Table 5. Radial Basis Function

Actual	1	2	3	4	5	6	7
1	67	0	7	0	1	2	2
2	0	58	0	5	6	7	1
3	16	4	79	1	10	5	15
4	0	24	2	67	9	1	4
5	3	6	2	20	56	13	8
6	2	3	1	2	6	64	4
7	12	5	9	5	12	8	66

$E_{rms} = 0.2607$ $Error_{\text{class}} = 35.6\%$

Table 6. Nearest Cluster

Actual	1	2	3	4	5	6	7
1	72	1	11	0	2	4	5
2	1	61	0	26	8	4	0
3	19	2	78	0	10	1	11
4	0	17	0	64	13	0	0
5	0	15	0	9	56	58	9
6	0	0	1	0	0	28	3
7	8	4	10	1	11	4	72

$E_{rms} = 0.263$ $Error_{\text{class}} = 37.5\%$

Table 7. Backpropagation Neural Network

Actual	1	2	3	4	5	6	7
1	75	0	1	0	1	2	4
2	0	83	0	0	2	1	0
3	11	2	94	1	1	0	8
4	0	7	1	96	8	0	0
5	6	8	0	3	75	5	6
6	1	0	0	0	5	84	3
7	8	0	4	0	8	8	79

$E_{rms} = 0.195$ $Error_{\text{class}} = 18.6\%$

7 Conclusions

The classifier architectures that gave suboptimal performance include the linear regression and unimodal Gaussian classifiers, both of which are parametric, and the radial basis function and nearest cluster classifier, which are nonparametric. One would expect the linear regression architecture to perform poorly; the generated hyperplanes can only be optimal in a Bayes sense if the feature distributions can be characterized as normal with equal covariance. Regardless, the simplicity of the linear regression algorithm is often favorable and has been a choice in the past [9] for feature distributions that are nonnormal.

The unimodal Gaussian suffers similar setbacks; the feature space is modeled as multivariate normal where each class is characterized by a mean vector and covariance matrix. Both the large E_{rms} (as in equation 18) and the percent error demonstrate its ineffectiveness. As we have stated previously [1,13], a multimodal classifier is more appropriate.

The RBF classifier also performed poorly, but, intuitively, we would expect that its flexibility in modeling non-Gaussian multimode distributions [8] would assist in classifier performance. On average, we see a low percentage of correct IDs but without the drastic errors in correct ID that other classifiers exhibit. The lower E_{rms} indicates higher confidence in the RBF decisions.

Past experience using *k*-means analysis suggested that the nearest cluster (NC) classifier might be beneficial. When exploring the dendrograms of hierarchical *k*-means clustering, we observed perhaps 3 to 5 clusters for each class of vehicle. In a limited sense, these clusters were used to classify the vehicles based on a simple distance metric. Here the NC method goes a step further in generating the posterior probabilities for class membership with respect to each cluster and input vector. However, the data-sets are more extensive and apparently there is great overlap in certain classes as can be seen by the class 5 and class 6 results. Therefore, overall performance is poor.

Acceptable classification performance was demonstrated by the Gaussian mixture, logistic regression, and BPNN classifier architectures. The Gaussian mixture classifier estimates the conditional probability distribution function using a weighted average of Gaussians, which provides an advantage due to the ability to approximate arbitrary distributions. The identification results are favorable, but we still observe a low confidence in this architecture's decision. This can be seen by the higher rms error results. This result is not totally unexpected, despite its advantages; the problem of estimating complex multimodal distributions still exists. The weighted summation does not completely address this problem.

For all its simplicity, the logistic regression architecture performed quite well. What can account for this? This parametric model, in effect, closely matches the underlying statistics primarily because it creates hyperplane boundaries similar to the true distributions. Also, in this case, a weighted sum of inputs is passed through a sigmoid nonlinearity to generate the output, which is very similar to more complex neural network architectures. Does this suggest that the underlying distribution is sigmoid in nature? Perhaps—certainly the underlying sigmoid kernel function is a better fit to the region of interest and explains why the Gaussian kernel, more suitable for radially centered data, does not perform as well. The logistic regression architecture is also considered to be a boundary decision classifier that may improve performance for multimodal distributions.

Finally, we see that the BPNN gives excellent results. Obviously, there are many similarities between the BPNN and the logistic regression classifier. The BPNN has the added advantage of the hidden layer, allowing for a finer parsing of the hyperspace and thus finer boundaries in the classification scheme.

To conclude, when one accounts for the training and testing time and the amount of memory required for the classifier, the logistic classifier architecture is a good choice for classification using this feature space. Its performance is exceptional at very low system cost.

Acknowledgments

This work was performed in-house by the Sensors and Electron Devices Directorate's Acoustic Signal Processing Branch at ARL in Adelphi, Maryland. The author would like to acknowledge the beneficial discussions and support of Nassy Srour, Doug Lake, and John Eicke.

References

[1] M. Wellman, N. Srour, and D. Hillis, ”Acoustic feature extraction for a neural network classifier,” Army Research Laboratory, ARL-TR-1166 (January 1997).

[2] N. Srour and J. Robertson, “Remote netted acoustic detection system: Final report,” Army Research Laboratory, ARL-TR-706 (April 1995).

[3] J. A. Robertson and B. Weber, “Artificial neural networks for acoustic target recognition,” Joint report between ARL and ITT Research.

[4] K. Eom, R. Chellappa, M. Wellman, N. Srour, and D. Hillis, “Acoustic target classification using multiscale methods,” *U.S. ARL 1997 Sensors and Electron Devices Symposium Proceedings* (January 1997).

[5] N. Berg, N. Srour, and G. Prado, “Unattended ground sensor performance, prediction and preliminary system requirements,” 1993.

[6] M. Wellman and N. Srour, “Feature extraction for a neural network classifier,” *3rd Joint Meeting of Acoustical Society of America and Acoustical Society of Japan* (December 1996).

[7] F. B. Shin and D. H. Kil, *Pattern Recognition and Prediction with Applications to Signal Characterization*, American Institute of Physics, Woodbury, New York (1996).

[8] S. Haykin, *Neural Networks, A Comprehensive Foundation*, Macmillan Publishing Company, New Jersey (1994).

[9] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, San Diego (1990).

[10] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, Inc. (1973).

[11] R. Kennedy, Y. Lee, C. Reed, and B. Roy, *Solving Pattern Recognition Problems*, Unica Technologies (1996).

[12] Database Mining Workstation, Reference Manual, HNC Inc. (1992).

[13] M. Wellman, N. Srour, and D. Hillis, “Feature extraction and fusion of acoustic and seismic sensors for target identification,” SPIE Aerosense Conference (April 1997).

[14] M. Wellman and N. Srour, “Enhanced target identification using higher order shape statistics,” Army Research Laboratory, ARL-TR-1723 (February 1999).

[15] D. Lake, “Tracking fundamental frequencies and least squares estimation of harmonic amplitude and phases,” Battlefield Acoustic Symposium (September 1997).

[16] K. Ng and R. Lippmann, “Practical characteristics of neural network and conventional pattern classifiers,” *Advances in Neural Information Processing Systems*, San Mateo, CA (1991).

[17] M. Powell, “Radial basis function approximation to polynomials,” *Numerical Analysis Proceedings*, Dundee, U.K. (1987).

[18] S. Chen, C. Cowan, and P. Grant, “Orthogonal least squares learning algorithm for radial basis function networks,” *IEEE Trans. Neural Networks*, **2**, No. 2 (March 1991).

[19] Q. Li and D. Tufts, “Discriminant networks: A simple, effective and rapidly trainable class of neural networks,” *IEEE Trans. Neural Networks* (February 1994).

[20] D. Kleinman, L. Kupper, and K. Muller, *Applied Regression Analysis and Other Multivariable Methods*, 2nd ed., PWS-Kent Publishing Company, Boston, MA (1988).